WE CLAIM:

1. A computer program product for controlling a computer to execute a computer program within a computer memory, said computer program product comprising:

(a) a loader program; and

(b) an encrypted version of said computer program; wherein

said loader program is operable to:

(i) read said encrypted version of said computer program stored in a program store;

(ii) decrypt said encrypted version of said computer program to form said computer program in an executable form;

(iii) load said computer program directly into said computer memory; and

(iv) trigger execution of said computer program as loaded into said computer memory by said loader program.

2. A computer program product as claimed in claim 1, wherein said encrypted version of said computer program is encrypted with a private encryption key and said loader program is operable to decrypt said encrypted version of said computer program with a corresponding public key.

3. A computer program product as claimed in claim 1, wherein said encrypted version of said computer program and said loader program are stored as separate computer files within a computer file store.

4. A computer program product as claimed in claim 1, wherein said loader program is associated with initialisation data specifying one or more of:

a storage location of said encrypted version of said computer program;

a key to be used in decrypting said encrypted version of said computer program; and

parameters specifying how said computer program should be loaded into said computer memory for execution.

5. A computer program product as claimed in claim 1, wherein said computer program is a malware scanning computer program.

6. A computer program product as claimed in claim 5, wherein said malware scanning computer program is operable such that once executing said malware scanning computer program scans said loader program for malware.

7. A computer program product as claimed in claim 6, wherein, if said loader program is detected as being infected with malware, then said malware scanning computer program is operable to repair said loader program.

8. A computer program product as claimed in claim 5, wherein said malware scanning computer program is operable to scan for malware including one or more of a computer virus, a worm, a Trojan, a banned computer file, a banned word and a banned image.

9. A computer program product as claimed in claim 1, wherein said loader program is operable to terminate after triggering execution of said computer program.

10. A computer program product as claimed in claim 1, wherein said computer program is operable to terminate said loader program when said computer program is triggered to execute by said loader program.

11. A computer program product as claimed in claim 1, wherein said loader program is operable to load said computer program into a memory space within said computer memory separate from a memory space used by said loader program.

12. A computer program product as claimed in claim 1, wherein said loader program is operable to load said computer program into an execution stream separate from an execution stream used by said loader program.

13. A method of executing of a computer program within a computer memory, said method comprising the steps of:

(a) executing a loader program, said loader program operating to:

(i) read an encrypted version of said computer program stored in a program store;

(ii) decrypt said encrypted version of said computer program to form said computer program in an executable form;

(iii) load said computer program directly into said computer memory; and

(iv) trigger execution of said computer program; and

(b) executing said computer program as loaded into said computer memory by said loader program.

14. A method as claimed in claim 13, wherein said encrypted version of said computer program is encrypted with a private encryption key and said loader program decrypts said encrypted version of said computer program with a corresponding public key.

15. A method as claimed in claim 13, wherein said encrypted version of said computer program and said loader program are stored as separate computer files within a computer file store.

16. A method as claimed in claim 13, wherein said loader program is associated with initialisation data specifying one or more of:

a storage location of said encrypted version of said computer program;

a key to be used in decrypting said encrypted version of said computer program; and

parameters specifying how said computer program should be loaded into said computer memory for execution.

17. A method as claimed in claim 13, wherein said computer program is a malware scanning computer program.

18. A method as claimed in claim 17, wherein once executing said malware scanning computer program scans said loader program for malware.

19. A method as claimed in claim 18, wherein, if said loader program is detected as being infected with malware, then said malware scanning computer program repairs said loader program.

20. A method as claimed in claim 17, wherein said malware scanning computer program scans for malware including one or more of a computer virus, a worm, a Trojan, a banned computer file, a banned word and a banned image.

21. A method as claimed in claim 13, wherein said loader program terminates after triggering execution of said computer programs.

22. A method as claimed in claim 13, wherein said computer program terminates said loader program when said computer program is triggered to execute by said loader program.

23. A method as claimed in claim 13, wherein said loader program loads said computer program into a memory space within said computer memory separate from a memory space used by said loader program.

24. A method as claimed in claim 13, wherein said loader program loads said computer program into an execution stream separate from an execution stream used by said loader program.

25. Apparatus for executing a computer program within a computer memory, said apparatus comprising:

(a) loader program logic; and

(b) a program store operable to store an encrypted version of said computer program; wherein

said loader program logic is operable to:

(i) read said encrypted version of said computer program stored in said program store;

(ii) decrypt said encrypted version of said computer program to form said computer program in an executable form;

(iii) load said computer program directly into said computer memory; and

(iv) trigger execution of said computer program as loaded into said computer memory by said loader program.

26. Apparatus as claimed in claim 25, wherein said encrypted version of said computer program is encrypted with a private encryption key and said loader program logic is operable to decrypt said encrypted version of said computer program with a corresponding public key.

27. Apparatus as claimed in claim 25, wherein said encrypted version of said computer program and said loader program are stored as separate computer files within a computer file store.

28. Apparatus as claimed in claim 25, wherein said loader program logic is associated with initialisation data specifying one or more of:

a storage location of said encrypted version of said computer program;

a key to be used in decrypting said encrypted version of said computer program; and

parameters specifying how said computer program should be loaded into said computer memory for execution.

29. Apparatus as claimed in claim 25, wherein said computer program is a malware scanning computer program.

30. Apparatus as claimed in claim 29, wherein said malware scanning computer program is operable such that once executing said malware scanning computer program scans said loader program logic for malware.

31. Apparatus as claimed in claim 30, wherein, if said loader program logic is detected as being infected with malware, then said malware scanning computer program is operable to repair said loader program logic.

32. Apparatus as claimed in claim 29, wherein said malware scanning computer program is operable to scan for malware including one or more of a computer virus, a worm, a Trojan, a banned computer file, a banned word and a banned image.

33. Apparatus as claimed in claim 25, wherein said loader program logic is operable to terminate after triggering execution of said computer programs.

34. Apparatus as claimed in claim 25, wherein said computer program logic is operable to terminate said loader program when said computer program is triggered to execute by said loader program.

35. Apparatus as claimed in claim 25, wherein said loader program logic is operable to load said computer program into a memory space within said computer memory separate from a memory space used by said loader program logic.

36. Apparatus as claimed in claim 25, wherein said loader program logic is operable to load said computer program into an execution stream separate from an execution stream used by said loader program logic.